科学技術計算用関数ライブラリ C-SSL II/VP の紹介

渡部 善隆 *

1999 年 1 月 5 日より C 言語および C++ 言語から利用できる科学技術計算用関数ライブラリ C-SSL II/VP(正式名称:UXP/V C-SSL II/VP V10) をスーパーコンピュータ VPP700/56 上で公開します。本稿では C-SSL II/VP の特徴、具体的な利用方法、オンラインマニュアルの参照方法について説明します。

センターでは連立 1 次方程式、固有値問題、乱数発生などの汎用的な機能を持つ Fortran サブルーチンライブラリとして SSL II/VPP([6]), SSL II/VP([3],[4],[5]) NUMPAC([7]) を公開しています。これらの Fortran ライブラリ中で SSL II/VP は C 言語からも直接呼び出すことができます ([2])。しかし利用の際には配列の格納方法に注意を払わなければならず、場合によっては行列を転置してサブルーチンに渡したりする必要もあります。

C-SSL II/VP は Fortran の SSL II/VP の中から使用頻度の高いものを自然なインターフェイスで利用できる C 言語の関数の形で提供しています。配列の格納順序の変更も不要になりました。プログラミングと実行可能ファイルの作成手順に少々 “くせ” があるものの、慣れてしまえば数値計算の強力な道具の一つになることと思います。なお本稿では VPP700/56 についての基礎的な知識を仮定しています。 VPP700/56 の利用方法の詳細は [1] を御覧ください。

1 C-SSL II/VP の概要

1.1 特徴

C-SSL II/VP はスーパーコンピュータ VPP700/56 (ホスト名: kyu-vpp, IP アドレス: 133.5.9.70) の単一 PE 上で動作する科学技術計算用関数ライブラリです。 C 言語および C++ 言語から利用できます。 C-SSL II/VP は倍精度実数型 (`double`) 変数を扱う関数 103 個と単精度実数型 (`float`) 変数を扱う関数 3 個から構成されています[1]。連立 1 次方程式・最小二乗解・固有値問題・代数方程式・ Fourier 変換・数値積分・補間・乱数などの汎用的な計算手続きをベクトル計算機向きにチューニングした関数として提供しています[2]。関数の一覧は 4 章を御覧ください。なお，複素数を扱う場合には専用の複素数型 “`dcomplex`” を定義します。

C-SSL II/VP は Fortran のサブルーチンライブラリである SSL II/VP を内部で呼び出します。そのため Fortran と C との結合規則に従った記述が必要になります。具体的な利用方法は 2 章で説明します。

関数と利用者プログラムとのデータ受渡しはすべて引数経由です。標準的な値が設定できる引数もあります。また、処理内容を整数値として返却する引数 (復帰値) はデバッグ、手法の有効性の検証データに活用することができます[3]。

*九州大学大型計算機センター・研究開発部; E-mail: `watanabe@cc.kyushu-u.ac.jp`

[1] 1998 年 12 月現在。

[2] 残念ながらソースプログラムは非公開です。従ってブラックボックスとなることを恐れて使うのをためらう方もおられると思います。その場合は自前のプログラムとの精度・速度の比較という目的で利用されてはいかがでしょうか。同じ問題を異なる手法・プログラムで実行、比較することは数値計算の信頼性を高めるための常道です。

[3] コンディションコードとも呼びます。慣れないうちは面倒なパラメータのような気がするかもしれません。しかし特異性の強い「きわどい」計算を行なう時には重要な役割を担うことがあります。

1.2 マニュアル

C-SSL II/VP のマニュアル [8](日本語), [9](英語) はセンター 4 階の図書室、2 階のプログラム相談室で見ることができます。その他、オンラインマニュアルとして PDF ファイルを提供しています。オンラインマニュアルの読み方は 3 章で説明します。

[8], [9] およびオンラインマニュアルには関数に用いた手法の詳しい解説は載っていません。SSL II/VP の手引書である [3], [4], [5] を参照してください。

2 C-SSL II/VP の利用方法

ここでは C-SSL II/VP を利用する際の一般的な注意事項、翻訳・結合によって実行可能ファイルを作成するまでの手順を説明します。なお、翻訳・結合による実行可能ファイルの作成、実行はもちろんバッチ処理でも行なうことができます。バッチ処理を行なう場合のシェルスクリプトの書き方、バッチリクエストの投入方法、処理コマンドのオプションの詳細は [1] を御覧ください。

2.1 C プログラムからの利用

2.1.1 注意事項

○標準ヘッダファイル

ソースプログラムの始めに必ず C-SSL II/VP の標準ヘッダファイル "`cssl.h`" をインクルードしてください。

```
#define "cssl.h"
```

`cssl.h` は `/usr/lang/cssl2vp/include/` の下にあり、中をのぞくことができます。`cssl.h` では複素数型 `dcomplex`、関数・引数の型などが宣言[4]されています。

○メイン関数名

メイン関数の関数名は `main` ではなく "`MAIN__`" とします。大文字の `MAIN` のあとにアンダースコアが 2 本です[5]。

○引数

データの受渡しは引数で行います。一般的な C 言語の受渡し方法と同じように、入力のみのスカラー変数は値渡しで、入出力または出力を行なうスカラー変数は変数へのポインタを渡します。

○行列

m 行 n 列の行列はプログラム内で "`a[m][n]`" と宣言し、i 行 j 列の要素 a_{ij} は "`a[i-1][j-1]`" に格納します。行列の添字は 1 から始まるのに C 言語の配列は 0 から始まることに注意してください。1 次元配列も同様です。ベクトルの要素 x_i は配列の要素 `x[i-1]` に対応します。

また行列を C-SSL II/VP の関数に引数として渡す場合は倍精度のポインタに型変換して渡します。今の例では "`(double*)a`" と書きます。また、`a` の列の数[6]を引数として渡す必要があります。

[4] プロトタイプ宣言とよばれます。

[5] これは Fortran と C の結合規則に従う必要性からきています。

[6] Fortran では「整合寸法」と呼びます。

○利用者関数

利用者関数は C 言語の一般的な関数として作成します。 C-SSLII/VP には関数名を渡します。

○複素数

複素数は標準ヘッダファイル “cssl.h” 中に dcomplex という名前で

```
      typedef struct {
         double re, im;
      } dcomplex;
```

と宣言されています。 “re” が実部、 “im” が虚部に対応する構造体です。以下のプログラムは倍精度複素数 za の宣言と値の設定例です。

```
#define NMAX 1000
MAIN__() {
      ....
dcomplex za[NMAX][NMAX]
    n = NMAX;
    for (i=0;i<n;i++)
      for (j=0;j<=i;j++) {
        za[i][j].re = i-n;
        za[j][i].re = n-i;
        za[i][j].im = 0;
        za[j][i].im = 0;
    }
       ....
```

○戻り値

すべての C-SSL II/VP の関数は long 型の値を返します。

0 : 正常終了
1 : 計算の途中でエラーが発生したか正しい計算結果が得られなかった

ことを意味します。もし関数の戻り値が “1” の場合、処理内容の戻り値として指定した変数の値を調べることでより詳細な情報を得ることができます。

2.1.2 サンプルプログラム

[8] から実行列と実ベクトルの積 c_dmav と実行列の連立 1 次方程式 c_dvlax を使用したプログラムを引用します。

```
#include <stdlib.h>
#include <stdio.h>
#include <math.h>
#include "cssl.h"         /* 標準ヘッダファイルのインクルード */

#define NMAX 100          /* 定数は #define で定義すると便利 */

MAIN__()                  /* 大文字 MAIN の後に 2 本のアンダースコア */
{
   long ierr, icon;
   long n, i, j, k, isw, is;
   double epsz, eps;
   double a[NMAX][NMAX], b[NMAX], x[NMAX], vw[NMAX];
   long ip[NMAX];

/* 行列 A と x を定義 */
    n = NMAX;
    for (i=0; i<n; i++) {
      for (j=i; j<n; j++) {
         a[i][j] = n-j;
         a[j][i] = n-j;
      }
    }
    for (i=0;i<n;i++) {
      x[i] = i+1;
    }

/* c_dmav によって b= Ax を計算; x は丸め誤差を無視すれば真の解になる */
/* 型の変換 (double*)a に注意 */
    k = NMAX;
    ierr = c_dmav((double*)a, k, n, n, x, b, &icon);

/* c_dvlax によって連立 1 次方程式を解く */
/* 型の変換 (double*)a に注意; 出力のあるスカラー変数 is, icon はポインタで渡す */
    epsz = 1e-12;
    isw = 1;
    ierr = c_dvlax((double*)a, k, n, b, epsz, isw, &is, vw, ip, &icon);

/* 戻り値が 0 でなかった場合の対応 */
    if (icon != 0) {
       printf("ERROR: c_dvlax failed with icon = %d\n", icon);
       exit(1);
    }

/* 精度の確認 */
    eps = 1e-6;
    for (i=0;i<n;i++) {
       if (fabs((x[i]-b[i])/b[i]) > eps) {
          printf("WARNING: result inaccurate\n");
          exit(1);
       }
    }
    printf("Result OK\n");
    return(0);
}
```

2.1.3 実行可能ファイルの作り方

○翻訳

翻訳は vcc(/usr/lang/bin/vcc) コマンドで行ないます。その際 -c オプションを指定してオブジェクトファイルのみ作成します。また -I オプションによって C-SSL II/VP の標準ヘッダーファイルのあるディレクトリ/usr/lang/cssl2vp/include を指定します。例ではプログラム名を “sample.c” としています。なお「 ↵ 」はリターンキーを押し下げることを意味します。

```
kyu-vpp% vcc -c -I/usr/lang/cssl2vp/include sample.c ↵      ←— 翻訳
```

翻訳が正常に終了すればオブジェクトファイル sample.o が作成されます。

○結合

次にオブジェクトファイルを C-SSL II/VP ライブラリと結合し、実行可能ファイルを作成します。作成には frt(/usr/lang/bin/frt) コマンドを用います。またオプションとして “-lcssl2vp -lssl2vp -lcvp” を指定します[7]。オプションの順番は入れ換えないでください。

```
kyu-vpp% frt sample.o -lcssl2vp -lssl2vp -lcvp ↵      ←— 実行可能ファイルの作成
```

結合処理が無事に終ると、実行可能ファイル a.out が作成されます。ファイル名を任意の名前に変更するには -o オプションに続けてファイル名を入力します。以下は実行可能ファイル名を “example.ex” に変更する例です。

```
kyu-vpp% frt -o example.ex sample.o -lcssl2vp -lssl2vp -lcvp ↵
```

2.1.4 単精度型を利用するには

単精度型関数や C-SSL II/VP がサポートしていない SSL II/VP の手続きを利用する場合は、直接 C プログラムから Fortran サブルーチンである SSL II/VP を呼び出します。利用方法は [2] を御覧ください。記事中の計算機 “VP2600/10” を “VPP700/56” と読み換えることでそのまま利用できます。

2.2 C++ プログラムからの利用

C++ プログラムから C-SSL II/VP を利用する場合には、以下の点に注意してください。

2.2.1 リンケージ指定

C++ プログラムでは呼び出す関数を extern "C"で宣言する必要があります。引数の並びは標準ヘッダーファイル/usr/lang/cssl2vp/include/cssl.h を見ながら作成してください。cssl.h はインクルードしません . 以下は関数 c_dvlax を宣言する例です。

```
extern "C" {
   long c_dvlax (  double a[], long k, long n, double b[], double epsz,
                   long isw, long *is, double vw[], long ip[], long *icon);
}
```

[7] 順番に C-SSL　II/VP ライブラリ (libcssl2vp.a) , SSL　II/VP ライブラリ (libssl2vp.a) , C/VP ライブラリ (libcvp.a) を呼び出しています。

2.2.2 実行可能ファイルの作り方

翻訳は CC(/usr/lang/bin/CC) コマンドで行ないます。結合は C 言語と同様 frt コマンドです。結合オプションとして C++ ライブラリを呼び出すオプション -lC を追加します。さらに実行可能プログラムに対して postld(/usr/lang/bin/postld) コマンドを実行します。

```
kyu-vpp% CC -c sample.c  ↵                                ←— 翻訳
kyu-vpp% frt sample.o -lcssl2vp -lssl2vp -lcvp -lC  ↵     ←— 実行可能ファイルの作成
kyu-vpp% postld a.out  ↵
```

2.2.3 C++ から使用できない関数

C-SSL II/VP では複素数型を C 言語の構造体を用いて実現しています。しかし C++ 言語と C 言語の結合では関数の引数として構造体を使用することができません。従って現在のところ複素数型を使用する関数は C++ プログラムから呼び出すことができません。

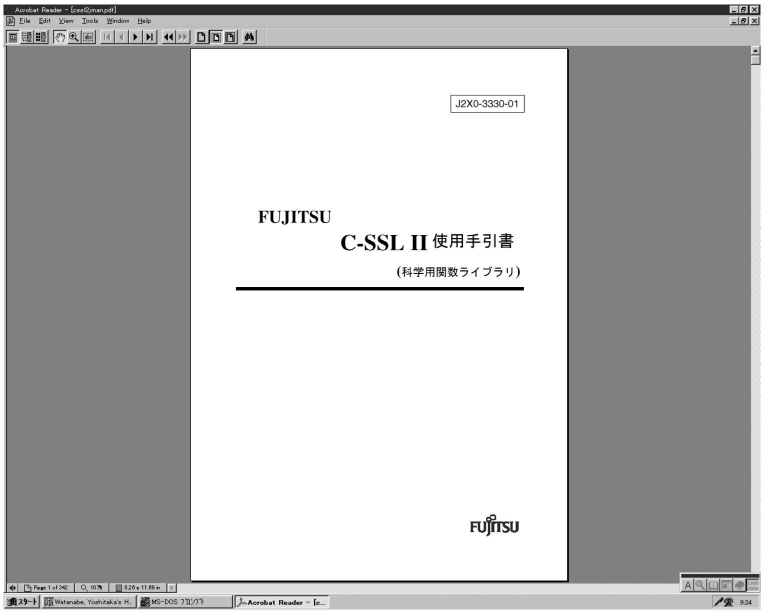
3 オンラインマニュアル

3.1 PDF ファイル

C-SSL II/VP のオンラインマニュアルは Adobe System 社の PDF(Portable Document Format) 形式です[8]。 kyu-vpp の次の場所にあります。

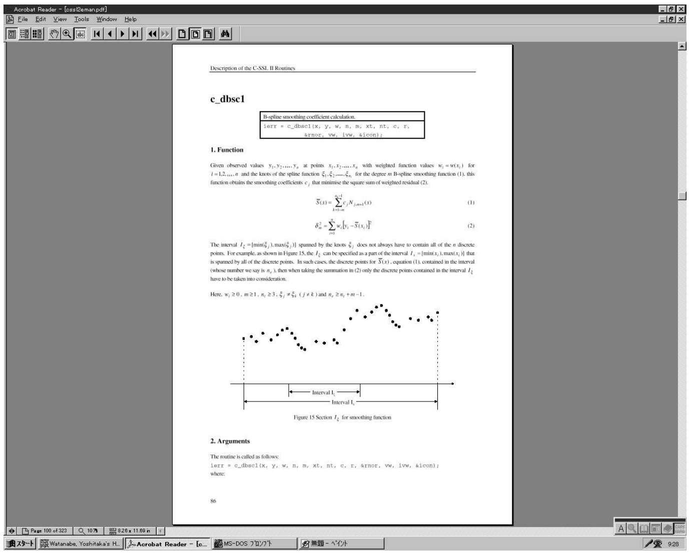
日本語版 (1,192,355 バイト): /usr/lang/cssl2vp/pdf/japan/cssl2jman.pdf
英語版 (1,114,011 バイト): /usr/lang/cssl2vp/pdf/C/cssl2eman.pdf

PDF ファイルは Adobe System 社から無料で提供されているツール "Acrobat Reader" を用いて読むことができます。バージョンは日本語の場合 3.0J 以降、英語の場合 3.0 以降が必要です。



C-SSL II/VP オンラインマニュアル (日本語版)

[8]内容に数式が多く含まれるため、 HTML では提供できなかったそうです。おそらく TEX で作成したものを PostScript 形式に変換し、ps2pdf で PDF に直したあとで修正を加えたものと思われます。



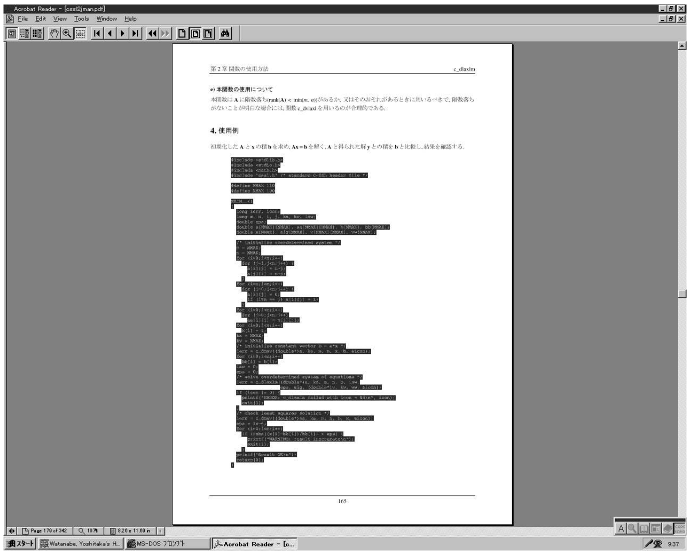
C-SSL II/VP オンラインマニュアル (英語版)

3.2 Acrobat Reader による検索

kyu-vpp で公開している C-SSL II/VP の PDF ファイルはセンター利用者なら誰でも手元のコンピュータに転送して読むことができます。転送は `ftp` コマンドなどを用いて バイナリーモード で行なってください。

転送したコンピュータに Acrobat Reader がインストールされてない場合は `http://www.adobe.co.jp/` からダウンロードします[9]。 Acrobat Reader の利用方法は Help メニューを参照してください。ここでは C-SSL II/VP に関連した注意事項をあげます。

- オンラインマニュアルの最初の一覧表の関数名をクリックするとその関数のページに移ることができます。
- テキスト選択ツールを使うことで、オンラインマニュアルのサンプルプログラムをコピーして他のツールに貼りつける (ペースト) ことができます。



テキストの選択．範囲を指定した後，右ボタンをクリックして copy を選択．

[9]Windows 95, Macintosh へのインストールは実に簡単です。

- Acrobat Reader の検索機能を用いたキーワード検索を行なうことができます。
- オンラインマニュアルを印刷することはできません。別途マニュアルを購入してください。

3.3 WWW ブラウザによる検索

センターのホームページ `http://www.cc.kyushu-u.ac.jp/` のオンラインマニュアルの項目から C-SSL II/VP の PDF ファイルへのリンクが張ってあります。Acrobat Reader がインストール済みのコンピュータであれば WWW ブラウザ用のプラグイン機能を用いてオンラインマニュアルを読むことができます。

ただし Acrobat Reader で利用できる検索機能やテキスト選択機能は使えません。プリンターへの出力もできません。また、センターホームページのオンラインマニュアルの利用はセンター課題をお持ちの方に限ります。

4 関数一覧

C-SSL II/VP で利用できる関数の一覧です。“(単精度)” の記述がないものはすべて倍精度実数または倍精度複素数を扱う関数です。

●行列格納モードの変換

c_dcgsm	行列格納モードの変換 (一般モード→対称行列)
c_dcsgm	行列格納モードの変換 (対称行列→一般モード)

●行列操作

c_daggm	行列の和 (実行列)
c_dsggm	行列の差 (実行列)
c_dvmggm	行列の積 (実行列)
c_dmav	実行列と実ベクトルの積
c_dmcv	複素行列と複素ベクトルの積
c_dvmvsd	スパース実行列と実ベクトルの積 (対角形式格納法)
c_dvmvse	スパース実行列と実ベクトルの積 (ELLPACK 形式格納法)

●行列の三角分解

c_dvalu	実行列の LU 分解 (ブロッキング LU 分解法)
c_dclu	複素行列の LU 分解 (Crout 法)

●連立 1 次方程式

c_dvlax	実行列の連立 1 次方程式 (ブロッキング LU 分解法)
c_dlcx	複素行列の連立 1 次方程式 (Crout 法)
c_dvlsx	正値対称行列の連立 1 次方程式 (変形 Cholesky 法)
c_dvlbx	実バンド行列の連立 1 次方程式 (Gauss の消去法)
c_dvlsbx	正値対称バンド行列の連立 1 次方程式 (変形 Cholesky 分解)
c_dlstx	正値対称 3 項行列の連立 1 次方程式 (変形 Cholesky 法)
c_dltx	実 3 項行列の連立 1 次方程式 (Gauss 消去法)
c_dvcgd	正値対称スパース行列の連立 1 次方程式 (前処理付き CG 法，対角形式格納法)
c_dvcge	正値対称スパース行列の連立 1 次方程式 (前処理付き CG 法， ELLPACK 形式格納法)
c_dvcrd	非対称または不定値スパース実行列の連立 1 次方程式 (MGCR 法，対角形式格納法)
c_dvcre	非対称または不定値スパース実行列の連立 1 次方程式 (MGCR 法， ELLPACK 形式格納法)

●逆行列

`c_dvluiv`	LU 分解された実行列の逆行列
`c_dcluiv`	LU 分解された複素行列の逆行列

●最小二乗解

`c_dlaxl`	実行列の最小二乗解 (Householder 変換)
`c_dlaxlm`	実行列の最小二乗最小ノルム解 (特異値分解法)
`c_dginv`	実行列の一般逆行列 (特異値分解法)
`c_dasvd1`	実行列の特異値分解 (Householder 法, QR 法)

●固有値，固有ベクトル

`c_deig1`	実行列の固有値及び固有ベクトル (2 段 QR 法)
`c_dceig2`	複素行列の固有値及び固有ベクトル (QR 法)
`c_dseig1`	実対称行列の固有値及び固有ベクトル (QL 法)
`c_dseig2`	実対称行列の固有値及び固有ベクトル (並列 Bisection 法, 逆反復法)
`c_dheig2`	Hermite 行列の固有値及び固有ベクトル (Householder 法, Bisection 法, 逆反復法)
`c_dbseg`	実対称バンド行列の固有値及び固有ベクトル (Lutishauser-Schwarz 法， Bisection 法，逆反復法)
`c_dteig2`	実対称 3 重対角行列の固有値及び固有ベクトル (Bisection 法, 逆反復法)
`c_dvgsg2`	実対称行列の一般固有値・固有ベクトル (並列 Bisection 法，逆反復法)

●非線形計算

`c_drqdr`	実係数 2 次方程式
`c_dcqdr`	複素係数 2 次方程式
`c_dlowp`	実係数低次代数方程式 (5 次以下)
`c_drjetr`	実係数高次代数方程式 (Jenkins-Traub の方法)
`c_dcjart`	複素係数高次代数方程式 (Jarratt 法)
`c_dtsd1`	実超越方程式 $f(x) = 0$ (Brent 法)
`c_dtsdm`	実超越方程式 $f(x) = 0$(Muller 法)
`c_dctsdm`	複素超越方程式 $f(z) = 0$(Muller 法)
`c_dnolbr`	連立非線形方程式 (Brent 法)

●補間

`c_daklag`	Aitken-Lagrange 補間
`c_dbif3`	B-spline 補間式 (III) による補間, 数値微分, 数値積分
`c_dbifd3`	B-spline2 次元補間式 (III-III) による補間, 数値微分, 数値積分
`c_dakmid`	2 次元準 Hermite 補間式による補間
`c_dakmin`	準 Hermite 補間式
`c_dbic3`	B-spline 補間式 (III)
`c_dbicd3`	B-spline2 次元補間式 (III-III)

●極値問題

`c_dlminf`	1 変数関数の極小化 (微係数不要, 2 次補間法)
`c_dminf1`	多変数関数の極小化 (微係数不要, 改訂準 Newton 法)
`c_dnolf1`	関数二乗和の極小化 (微係数不要, 改訂 Marquardt 法)
`c_dlprs1`	線形計画問題 (改訂 simplex 法)
`c_dnlpg1`	非線形計画問題 (微係数要, Powell 法)

●平滑化

`c_dsmle1`	最小二乗近似多項式による平滑化 (等間隔離散点)
`c_dbsf1`	B-spline 平滑化式による平滑化, 数値微分, 数値積分
`c_dbsc1`	B-spline 平滑化式 (固定節点)

●変換

c_dvmcft	1 次元，多重，多次元離散複素 Fourier 変換 (混合基底)
c_dvcft1	離散型複素 Fourier 変換 (2 基底 FFT)
c_dvrft1	離散型実 Fourier 変換 (2 基底 FFT)
c_dvcos1	離散型 cosine 変換 (2 基底 FFT)
c_dvsin1	離散型 sine 変換 (2 基底 FFT)

●近似

c_dlesq1	最小二乗近似多項式

●疑似乱数

c_dvrau4	一様乱数 $[0, 1)$ の生成
c_dvran3	正規乱数の生成
c_rane2	指数乱数の生成 (単精度)
c_ranp2	Poisson 乱数の生成 (単精度)
c_ranb2	二項乱数の生成 (単精度)

●特殊関数

c_dceli1	第 1 種完全楕円積分 $K(x)$
c_dceli2	第 2 種完全楕円積分 $E(x)$
c_dsini	正弦積分 $S_i(x)$
c_digam1	第 1 種不完全 Gamma 関数 $\gamma(\nu, x)$
c_digam2	第 2 種不完全 Gamma 関数 $\Gamma(\nu, x)$
c_dbj0	第 1 種 0 次 Bessel 関数 $J_0(x)$
c_dbj1	第 1 種 1 次 Bessel 関数 $J_1(x)$
c_dby0	第 2 種 0 次 Bessel 関数 $Y_0(x)$
c_dby1	第 2 種 1 次 Bessel 関数 $Y_1(x)$
c_dbi0	第 1 種 0 次変形 Bessel 関数 $I_0(x)$
c_dbi1	第 1 種 1 次変形 Bessel 関数 $I_1(x)$
c_dbk0	第 2 種 0 次変形 Bessel 関数 $K_0(x)$
c_dbk1	第 2 種 1 次変形 Bessel 関数 $K_1(x)$
c_dbjn	第 1 種整数次 Bessel 関数 $J_n(x)$
c_dbyn	第 2 種整数次 Bessel 関数 $Y_n(x)$
c_dbin	第 1 種整数次変形 Bessel 関数 $I_n(x)$
c_dbkn	第 2 種整数次変形 Bessel 関数 $K_n(x)$
c_dcbin	複素変数第 1 種整数次変形 Bessel 関数 $I_n(z)$
c_dcbkn	複素変数第 2 種整数次変形 Bessel 関数 $K_n(z)$
c_dcbjn	複素変数第 1 種整数次 Bessel 関数 $J_n(z)$
c_dcbyn	複素変数第 2 種整数次 Bessel 関数 $Y_n(z)$
c_dbjr	第 1 種実数次 Bessel 関数 $J_\nu(x)$
c_dbyr	第 2 種実数次 Bessel 関数 $Y_\nu(x)$
c_dbir	第 1 種実数次変形 Bessel 関数 $I_\nu(x)$
c_dbkr	第 2 種実数次変形 Bessel 関数 $K_\nu(x)$
c_dcbjr	複素変数第 1 種実数次 Bessel 関数 $J_\nu(z)$

●微分方程式

c_dodge	stiff 連立 1 階常微分方程式 (Gear 法)
c_dodam	連立 1 階常微分方程式 (Adams 法)

●数値積分

c_dsimp1	1 次元有限区間積分 (等間隔離散点入力, Simpson 則)
c_dtrap	1 次元有限区間積分 (不等間隔離散点入力, 台形則)
c_daqn9	1 次元有限区間積分 (関数入力, 適応型 Newton-Cotes9 点則)
c_daqe	1 次元有限区間積分 (関数入力, 二重指数関数型積分公式)
c_daqeh	1 次元半無限区間積分 (関数入力, 二重指数関数型積分公式)
c_daqei	1 次元全無限区間積分 (関数入力, 二重指数関数型積分公式)
c_daqmc8	多次元有限領域積分 (関数入力, Clenshaw-Curtis 型積分法)

参考文献

[1] VPP700/56 利用の手引 (第 2 版), 九州大学大型計算機センター, 1998.

[2] 渡部 善隆，山元 規靖: C プログラムから SSL II を利用するには, 九州大学大型計算機センター広報, Vol.29, No.3, pp.234–242 (1996).

[3] 富士通 SSL II 使用手引書 (科学用サブルーチンライブラリ), 99SP-4020, 富士通株式会社, 1987.

[4] FIJITSU SSL II 拡張機能使用手引書 (科学用サブルーチンライブラリ), 99SP-4070, 富士通株式会社, 1991．

[5] FUJITSU SSL II 拡張機能使用手引書 II(科学用サブルーチンライブラリ), J2X0-1360, 富士通株式会社, 1997．

[6] FUJITSU SSL II/VPP 使用手引書 (科学用サブルーチンライブラリ)V12 用, J2X0-1370, 富士通株式会社, 1997．

[7] NUMPAC 利用手引書, 富士通株式会社, 1994．(http://cronos.fuis.fukui-u.ac.jp/numpac/ から検索可能)

[8] FUJITSU C-SSL II 使用手引書 (科学用関数ライブラリ), J2X0-3330-01, 富士通株式会社, 1997.

[9] Fujitsu C-SSL II User's Guide, J2X0-3330-01EN, 富士通株式会社, 1997.